Lab 9

Objectives:
More practice designing classes and methods.
Practice creating graphical objects.

Useful Links:
- www.csc.villanova.edu/~map/1051/Chap04/Splat.java
- www.csc.villanova.edu/~map/1051/Chap04/SplatPanel.java
- www.csc.villanova.edu/~map/1051/Chap04/Circle.java
- www.csc.villanova.edu/~map/1051/Chap02/Snowman.java
- www.csc.villanova.edu/~map/1051/Chap04/Die.java
- www.csc.villanova.edu/~map/1051/Chap04/Account.java
- lecture slides: www.csc.villanova.edu/~map/1051/f15/09Classes2.pdf
- GUI class exercise: www.csc.villanova.edu/~map/1051/f15/09xGUIexercise.pdf

1. Starting with the Happiness, HappinessPanel, and Smiley classes from the GUI class exercise

a) Demonstrate your program and GUI Object exercise worksheet:

Instructor/TA initials: ______________

b) Modify the Smiley class further, making it more like the Person class – i.e., add name, age, and happiness state attributes.
You will need to also modify the way the constructor and the draw() methods work:
- Add another constructor with more parameters that allows you to set the name, age, and happiness to other values.
- Modify the old constructor (the one that only has color, x, and y as parameters): it should also set the name, age, and happiness to some default values, such as “Smiley”, 0, and true, respectively.
- Modify the draw() method so that the smiley looks different depending on the age and happiness state and to include the name (Hint: Use the drawString() method to display the name below the smiley).

Test your code well before proceeding and discuss your work with your partner. Write comments on your Lab partner’s worksheet and have him or her write on yours, below.

Lab Partner name: ____________________________________________

Lab partner comments: eg, what did you learn by sharing your work on this exercise with your partner?
2. A Snowman class

Create a **Snowman** class, similar to the **Smiley** class. You can use much of the code from the [Snowman applet](#) – the constants **MID** and **TOP** will now be your instance variables, corresponding to the x, y position, as in the **Smiley** class. (You should rename **MID** and **TOP** to x and y or to mid and top and declare them **private**). You don’t need any other instance variables, unless you plan to have different versions of the **Snowman** (e.g: arms up vs. arms down). Test your code by drawing a few Snowmen in your **HappinessPanel**.

3. A draw() method for the Die class

A die can be depicted by a white square outlined in black, with the number inside (or you can take the challenge and try to make it look like a real die, but drawing the little dots corresponding to the **faceValue** is tricky).

Note that the **Die** class does not have x and y (position) attributes. Rather than adding these attributes, we will take a different approach, and incorporate x, y as parameters to the **draw()** method that we are writing. Thus, in the **paintComponent()** method of **HappinessPanel**, we will use:

```java
die1.draw(page, 40, 60);
```

(instead of **die1.draw(page)**; which assumes **die1** has a position)

Therefore, the draw() method in the Die class will need a different heading:

```java
public void draw( Graphics page, int x, int y)
```

The method definition should use the position x, y to draw the die, as follows:

a. use **fillRect()** to create a white square (or use it twice to create a white square with a black outline)

b. use **drawString()** to put the String corresponding to **faceValue** inside the white square. Note: remember that **faceValue** is an int, so use **Integer.toString(faceValue)** to convert it to a String, as is done in the original **toString()** method of **Die** class.

c. Test your code by drawing a few dice in your **HappinessPanel**.

d. Test it again, inserting **die1.roll()** right before the draw() method is invoked in the **paintComponent()** method of **HappinessPanel**, for one of the dice. Observe the dice as you resize the window: note that the **paintComponent()** method is invoked every time the window is resized, so if a die is rolled in **paintComponent()**, you should see that die’s **faceValue** change as you resize.

**Demonstrate your work**

- Demonstrate how resizing the window alters the image: ______

**Files:**

- Happiness ______
- HappinessPanel ______
- Smiley ______
- Snowman ______
- Die ______